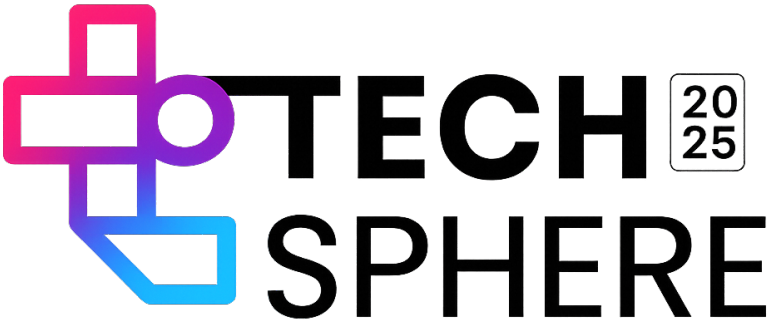
Challenge de Clasificación Biomédica con IA



Informe final del reto

Integrantes

Carlos Leonardo Bravo Revelo

Mateo Alejandro Bravo Revelo

Alexander Passo Polanco

Medellín, Antioquia

2025

Índice

1. Introducción

El Data + AI Challenge 2025, organizado por Tech Sphere Colombia, representa una oportunidad única para que profesionales emergentes en ciencia de datos demuestren su capacidad para resolver problemas reales mediante inteligencia artificial. Este desafío invita a construir una solución capaz de clasificar artículos médicos en dominios específicos como cardiovascular, neurológico, hepatorrenal y oncológico, utilizando únicamente el título y el resumen del artículo. Los participantes pueden emplear enfoques tradicionales de machine learning, modelos de lenguaje, flujos de trabajo con agentes de IA o soluciones híbridas, siempre que justifiquen y demuestren la efectividad de su elección tecnológica. [2]

La clasificación automática de literatura médica es un ejemplo claro de cómo el machine learning (ML) puede transformar sectores como la salud, facilitando la organización y acceso a información crucial para la investigación y la toma de decisiones clínicas. En este contexto, el ML permite entrenar modelos que aprenden patrones y relaciones dentro de grandes volúmenes de datos, mejorando la precisión y eficiencia en tareas complejas. Sin embargo, para que estos modelos sean efectivos, es esencial contar con datos de calidad, una adecuada preparación del dataset y una justificación técnica sólida para las decisiones tomadas durante el proceso de desarrollo. [1]

En el ámbito colombiano, iniciativas como el Data + AI Challenge son fundamentales para fortalecer el ecosistema de inteligencia artificial. Colombia ha avanzado en la creación de políticas públicas que promueven el uso ético y responsable de la IA, como la Política Nacional de Inteligencia Artificial aprobada en 2024, que establece acciones específicas para fomentar su desarrollo y adopción en diversos sectores . Eventos como este desafío no solo impulsan la innovación local, sino que también posicionan a Medellín como un hub emergente en ciencia y tecnología en América Latina . [1]

2. Análisis exploratorio y comprensión del problema

El análisis exploratorio se lo realizó en el siguiente notebook EDA. El **dataset challenge_data-18-ago** está compuesto por 3,566 registros, cada uno correspondiente a un artículo médico. Para cada artículo se dispone de dos fuentes textuales principales: el título y el abstract, los cuales contienen la información científica esencial del estudio. El objetivo del dataset es la clasificación automática de artículos

en el ámbito biomédico. Cada registro está etiquetado con una o más categorías pertenecientes a cuatro áreas clínicas principales:

- Cardiovascular
- Hepatorenal
- Neurological
- Oncological

La figura 1 presenta la distribución de artículos por grupo de clasificación médica. Se observa un claro desbalance en las clases: la categoría neurological es la más frecuente, superando ampliamente a las demás, con más de 1,000 artículos registrados. En segundo plano aparecen cardiovascular y hepatorenal, con una representación moderada, mientras que la clase oncological resulta menos común cuando aparece sola.
Asimismo, las combinaciones de clases, por ejemplo, neurological—cardiovascular, neurological—hepatorenal o hepatorenal—oncological— presentan frecuencias considerablemente más bajas, en la mayoría de los casos por debajo de los 200 artículos. Este patrón refleja que, aunque existen artículos asociados a más de un ámbito médico, su proporción dentro del conjunto de datos es reducida.

La distribución evidencia que el dataset está fuertemente desbalanceado, con predominio de la categoría neurological y una subrepresentación de oncological y de las combinaciones multilabel.

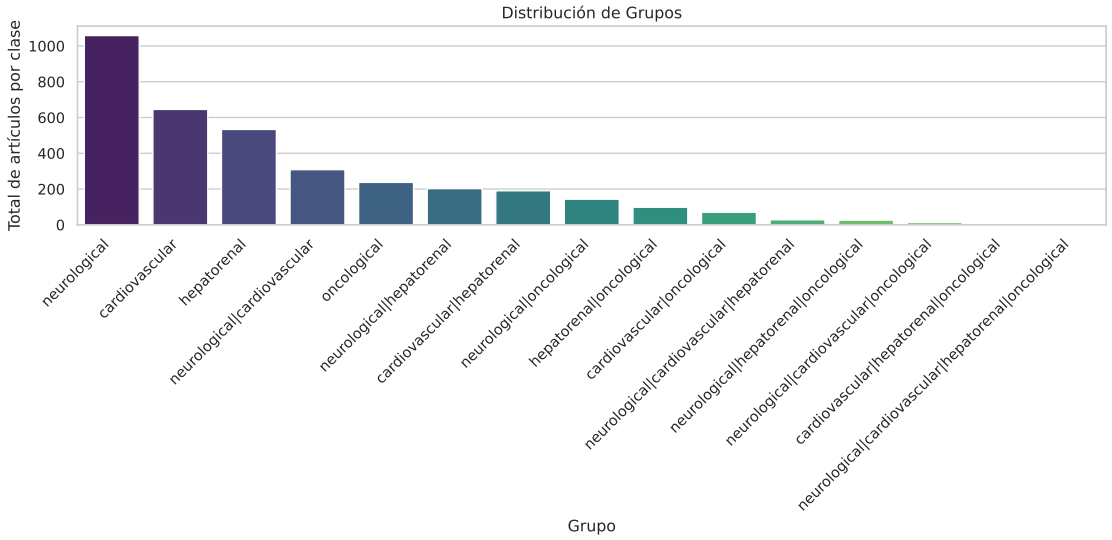


Figura 1: Distribución de grupos

A partir de la distribución de longitudes de los abstracts se observa un claro sesgo a la derecha (right-skewed), como se observa en la Figura 2. La mayoría de los resúmenes son relativamente cortos, concentrándose principalmente entre las 20 y 40 palabras, con un pico muy pronunciado alrededor de las 30 palabras, que parece ser la longitud más común.

Sin embargo, la distribución presenta una cola larga hacia la derecha, lo que indica que, aunque la mayoría de los abstracts son breves, existe un número considerable que alcanza entre 100 y 200 palabras. Incluso se identifican casos excepcionales que superan las 300 palabras, correspondientes a resúmenes notablemente extensos. (Figura 2

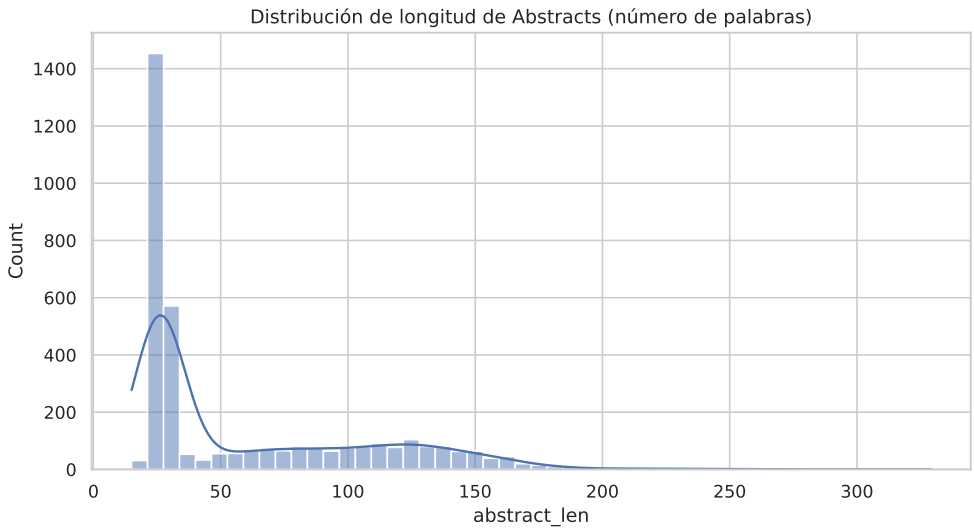


Figura 2: Distribución de longitud de Abstract

La figura 3 presenta la distribución de la longitud de los títulos de los artículos, medida en número de palabras. Se observa que la mayoría de títulos son breves y concisos, concentrándose principalmente entre 4 y 6 palabras, con un pico muy marcado en torno a las 5 palabras, lo cual corresponde al estilo típico de los artículos médicos.

La distribución tiene una cola hacia la derecha, indicando la presencia de algunos títulos más extensos que superan las 15 o 20 palabras, aunque estos casos son poco frecuentes y pueden considerarse excepcionales. En contraste, también se identifican títulos extremadamente cortos, de apenas una o dos palabras, posiblemente relacionados con abreviaturas o formatos atípicos.

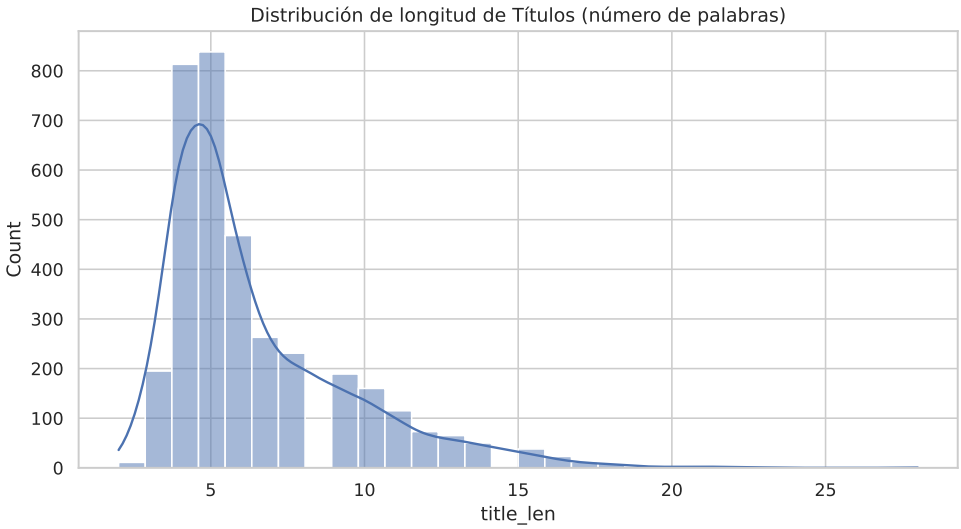


Figura 3: Distribución de longitud de Titulo

Estos resultados reflejan que los títulos, a diferencia de los abstracts, presentan una estructura más homogénea y limitada en extensión, lo que los hace menos informativos por sí solos, pero útiles como complemento del contenido del abstract dentro de un proceso de clasificación de textos biomédicos.

3. Preparación y preprocesamiento

3.1. Procesamiento de datos:

Todo el proceso de Procesamiento de Datos, entrenamiento, metricas se encuentra en el notebook Data_Challenge_Multilabel Primero, se cargan los datos desde un archivo CSV utilizando la función read_csv de la librería pandas. Esta lectura permite obtener un DataFrame con las columnas title, abstract y group, las cuales contienen respectivamente el título, el resumen y las etiquetas asociadas a cada artículo.

Posteriormente, se construye la columna de texto de entrada concatenando el título y el abstract en un único campo. Esta estrategia se utiliza porque el título ofrece un resumen conciso del contenido, mientras que el abstract proporciona información detallada sobre el contexto y los objetivos del artículo. Al combinarlos, se genera una representación textual más completa, lo que facilita que los modelos basados en BERT, como SciBERT, capturen de manera más efectiva tanto la información general.

En cuanto a las etiquetas, se procesan con el método split("—"), dividiendo cada cadena de categorías en una lista. Esto es necesario porque un mismo texto puede pertenecer a múltiples grupos (clasificación multilabel), y expresarlo en forma de lista facilita el procesamiento posterior.

Luego, estas listas de etiquetas son transformadas en vectores binarios mediante MultiLabelBinarizer, el cual convierte cada etiqueta en un valor 1 o 0 según esté presente o no en la muestra. Este paso es indispensable para que el modelo pueda trabajar con valores numéricos en lugar de texto.

Para asegurar la reutilización y consistencia del preprocesamiento de etiquetas, el codificador MultiLabelBinarizer se guarda en dos formatos: uno en binario mediante joblib y otro en JSON, que permite consultar de manera clara la lista de clases. Esta estrategia garantiza que, en cualquier predicción futura, los vectores binarios generados por el modelo puedan mapearse de forma correcta a sus etiquetas originales.

Finalmente, se divide el dataset en dos subconjuntos: entrenamiento (80 %) y prueba (20 %). Esta separación asegura que el modelo pueda aprender a partir de una porción de los datos y luego ser evaluado en ejemplos nuevos, permitiendo medir su capacidad de generalización.

3.2. Tokenización de textos:

La tokenización se realizan utilizando SciBERT, un modelo preentrenado específico para lenguaje científico. Se selecciona el modelo allenai-scibert-scivocab-uncased porque su vocabulario y embeddings están optimizados para textos científicos, lo que mejora la representación semántica frente a modelos BERT generales. El tokenizador convierte cada texto en una secuencia de tokens que el modelo puede procesar, aplicando truncamiento y padding para garantizar que todas las secuencias tengan la misma longitud (max-length=256). Esto es crucial porque los modelos basados en Transformers requieren que los inputs tengan la misma dimensión para poder construir los tensores de entrada correctamente.

Posteriormente, se define una clase personalizada. Esta clase encapsula los encodings generados por el tokenizador y las etiquetas binarias correspondientes a cada muestra. Implementar los métodos permite que PyTorch pueda acceder a los datos de manera eficiente durante el entrenamiento y la evaluación, extrayendo automáticamente los tensores de entrada (input-ids, attentionmask, etc.) y las etiquetas como tensores de tipo float.

Finalmente, se crean los objetos train_dataset y test-dataset, que combinan los encodings y las etiquetas de las muestras de entrenamiento y prueba. Esta estructura es fundamental para el entrenamiento eficiente con el Trainer de Hugging Face.

4. Selección y diseño de la solución

4.1. Selección de modelo:

El modelo seleccionado SciBERT, es una variante de BERT (Es un modelo que entiende el significado de las palabras dentro de su contexto completo usando la arquitectura Transformer, incluyendo clasificación multilabel) entrenada específicamente con textos científicos, lo que le permite capturar de manera efectiva el vocabulario técnico. Esta característica es fundamental para nuestro dataset, donde los términos especializados y el contexto médico-científico son predominantes. A diferencia de un BERT genérico, SciBERT posee embeddings preentrenados en un conjunto de datos de investigación.

El enfoque multilabel se ajusta a la naturaleza del problema, ya que un artículo puede pertenecer simultáneamente a varias categorías médicas. Configurar el modelo con `problem_type="multi_label_classification"` y `num_labels=len(mlb.classes_)` garantiza que la salida del modelo genere probabilidades independientes para cada etiqueta, permitiendo que múltiples clases se predigan simultáneamente. Esto es superior a enfoques de clasificación multiclase tradicional, donde solo se puede asignar una etiqueta por instancia, lo que sería insuficiente para capturar la complejidad de nuestro dataset.

En cuanto a la comparación con LLMs, si bien estos modelos ofrecen capacidades generativas y de razonamiento muy avanzadas, su uso directo para clasificación supervisada presenta limitaciones importantes: requieren prompts cuidadosamente diseñados, tienen costos computacionales más elevados y no garantizan consistencia en la salida de etiquetas. Además, la evaluación directa de métricas como F1-score y la construcción de una matriz de confusión es mucho más natural y reproducible con un modelo como SciBERT. Por estas razones, se priorizó un enfoque basado en fine-tuning de modelos preentrenados específicos del dominio.

4.2. Parámetros de entrenamiento

- **learning_rate** $= 2e-5$**:** A una tasa de aprendizaje alta, el modelo puede sobrepasar el mínimo en la función de pérdida y no converger adecuadamente.

Si es demasiado baja, el modelo podría converger demasiado lento, requiriendo muchas más épocas de entrenamiento y gastando más tiempo. $2e{-}5$ es una tasa de aprendizaje comúnmente utilizada para modelos grandes como SciBERT, ya que es suficientemente pequeña para evitar grandes oscilaciones.

- **per_device_train_batch_size = 8:** El tamaño del lote afecta la eficiencia del entrenamiento, si es muy pequeño, el modelo se actualiza con más frecuencia, pero la estimación del gradiente es más ruidosa. Si es muy grande, se requieren más recursos de memoria y puede ser más difícil encontrar el mínimo óptimo. Un tamaño de lote de 8 es una buena elección para empezar, y es especialmente útil si trabajas con GPUs con limitaciones de memoria.

- **num_train_epochs = 5:** Cuántas veces el modelo verá todo el conjunto de datos de entrenamiento, influye directamente en cuánto tiempo tomará el modelo para aprender. Demasiadas épocas pueden llevar a overfitting (sobreajuste), donde el modelo memoriza los datos en lugar de aprender patrones generales. Pocas épocas pueden resultar en un subajuste (underfitting), donde el modelo no aprende lo suficiente.

4.3. Resultado de proceso de entrenamiento:

Epochs	Training Loss	Validation Loss	F1 W	Precision W	Recall W	Hamm.Loss
1	No log	0.143081	0.9263	0.9600	0.9069	0.0483
2	0.2342	0.138344	0.9362	0.9601	0.9142	0.0413
3	0.0972	0.129111	0.9574	0.9500	0.9215	0.0110
4	0.0972	0.135501	0.9450	0.9518	0.9351	0.0364
5	0.0554	0.154343	0.9406	0.9350	0.9465	0.0403

Cuadro 1: Métricas de rendimiento a lo largo de las épocas

1. Training Loss: A medida que avanzan las épocas, esta métrica disminuye (0.2342 en la época 2 a 0.0554 en la época 5), lo que indica que el modelo está mejorando en términos de aprendizaje en los datos de entrenamiento.

2. Validation Loss: También disminuye ligeramente con el tiempo (de 0.143081 en la época 1 a 0.154343 en la época 5), lo que sugiere que el modelo está generalizando bien, aunque hay algo de variabilidad.

3. F1 W (F1 ponderado): Se mantiene relativamente constante en torno a 0.93, lo que muestra que el modelo tiene un buen desempeño en términos de precisión y recall combinados.

4. Precision W (Precisión ponderada): Esta métrica permanece muy estable en torno al 0.96, lo que indica que el modelo tiene una alta precisión en las predicciones positivas.

5. Recall W (Recall ponderado): Aunque el recall disminuye ligeramente de 0.9069 en la época 1 a 0.9351 en la época 4, se estabiliza alrededor de 0.935 en la época 5, lo que muestra que el modelo sigue siendo capaz de identificar correctamente las clases positivas.

6. Hamm. Loss (Pérdida de Hamming): Esta métrica disminuye notablemente con las épocas, lo que indica que el modelo mejora al reducir los errores de clasificación.

A partir de los datos presentados, parece que el modelo ha alcanzado su punto óptimo después de 4 o 5 épocas. Seguir entrenando más allá de este punto podría generar un sobreajuste, donde el modelo comienza a perder capacidad de generalización, y las mejoras en las métricas de entrenamiento no justifican el aumento en el tiempo de entrenamiento y los recursos computacionales necesarios

5. Métricas

5.1. F1 Score, Recall, Precisión y Exactitud:

En este 'challenge', se ha definido un conjunto de métricas para evaluar el desempeño del modelo de clasificación multilabel basado en SciBERT. Dada la naturaleza multilabel del problema, es fundamental considerar métricas que reflejen tanto la precisión global de las predicciones como el desempeño individual de cada clase, para así evitar sesgos hacia clases mayoritarias.

Se utilizan métricas de promedio micro (micro) y ponderado (weighted) para f1, precisión y recall. El promedio micro calcula las métricas considerando todas las etiquetas de manera acumulativa y luego las promedia. Por su parte, el promedio ponderado tiene en cuenta la proporción de cada clase, permitiendo evaluar si el modelo mantiene un desempeño equilibrado entre etiquetas frecuentes y más escasas.

Adicionalmente, se calcula la métrica de exact match ratio, que representa la proporción de muestras donde todas las etiquetas se predicen correctamente, proporcionando una medida estricta de exactitud en tareas multilabel. Complementariamente, la Hamming Loss mide la diferencia entre las predicciones de un modelo y las etiquetas reales, basándose en cuántos valores de etiquetas son incorrectos

La combinación de estas métricas permite tener una visión integral del desempeño del modelo, evaluando su capacidad de generalización, precisión, recall y exactitud total. Esto asegura que el modelo no solo aprenda a predecir correctamente la mayoría de etiquetas, sino que también mantenga consistencia y confiabilidad en nuevas predicciones.

5.2. Matrices de confusión:

En problemas de clasificación multilabel, cada muestra puede pertenecer a más de una clase simultáneamente, a diferencia de la clasificación tradicional multiclase donde cada muestra tiene una sola etiqueta. Por esta razón, para evaluar el desempeño del modelo es necesario calcular una matriz de confusión independiente por cada clase, en lugar de una única matriz global.

La matriz de confusión para cada clase muestra los siguientes valores:

- TN (True Negative): Número de muestras que no pertenecen a la clase y fueron correctamente clasificadas como negativas
- FP (False Positive): Número de muestras que no pertenecen a la clase pero fueron clasificadas incorrectamente como positivas.
- FN (False Negative): Número de muestras que pertenecen a la clase pero fueron clasificadas incorrectamente como negativas.
- TP (True Positive): Número de muestras que pertenecen a la clase y fueron correctamente clasificadas como positivas.

Estos cuatro valores (TN, FP, FN, TP) permiten calcular métricas fundamentales como precisión, recall, F1-score y Hamming Loss para cada etiqueta. Mostrar los cuatro componentes es crucial porque cada uno aporta información diferente sobre los errores y aciertos del modelo, permitiendo identificar si el modelo tiende a predecir en exceso ciertas clases (alto FP) o falla en detectar algunas etiquetas importantes (alto FN).

En el código presentado, se utiliza multilabel-confusion-matrix de scikit-learn, que devuelve un arreglo de matrices, una por cada clase del dataset. Luego, se visualiza cada matriz con ConfusionMatrixDisplay, permitiendo interpretar de manera individual el desempeño del modelo para cada etiqueta, lo que es especialmente útil en datasets multilabel con clases desequilibradas.

6. Resultados

En esta sección se presentan los resultados obtenidos tras el proceso de evaluación del modelo en la quinta época de entrenamiento. Las métricas reportadas incluyen medidas de desempeño clásicas como *Precision*, *Recall* y *F1-Score*, tanto en su forma estándar como ponderada, lo que permite analizar el balance entre falsos positivos y falsos negativos en diferentes clases. Asimismo, se incluyen métricas adicionales como la *Exact Match Ratio* y la *Hamming Loss*, que resultan especialmente útiles en escenarios de clasificación multietiqueta. Finalmente, se reportan también indicadores de eficiencia computacional como el tiempo de ejecución, el número de muestras procesadas por segundo y el número de pasos por segundo. La Tabla 2 resume de manera detallada estos valores, evidenciando un desempeño robusto del modelo.

Cuadro 2: Resultados de evaluación del modelo

Métrica	**Valor**
Loss de evaluación	0.1326
F1	0.9354
Precisión	0.9474
Recall	0.9236
F1 (ponderado)	0.9353
Precisión (ponderada)	0.9473
Recall (ponderado)	0.9236
Exact Match Ratio	0.8527
Hamming Loss	0.0428
Tiempo de evaluación (s)	10.05
Muestras por segundo	70.93
Steps por segundo	8.95
Época	5

Los resultados alcanzados muestran que el modelo tuvo un desempeño bastante sólido en la tarea de clasificación. El valor de F1-score obtenido ($\approx 0{,}93$) indica un buen equilibrio entre precisión y exhaustividad, lo que significa que el modelo no solo logra identificar correctamente la mayoría de los casos positivos, sino que también comete pocos errores al incluir ejemplos que no corresponden. En la misma línea, la precisión cercana al 0.95 refleja que, en general, las predicciones positivas hechas por el modelo son correctas, mientras que el recall ($\approx 0{,}92$) evidencia su capacidad para recuperar la mayoría de las instancias relevantes.

Por otra parte, la métrica de Exact Match Ratio ($\approx 0{,}85$) confirma que en un alto porcentaje de los casos el modelo logra predecir todas las etiquetas de manera exacta, y el bajo valor de Hamming Loss ($\approx 0{,}04$) respalda la idea de que los errores

en las etiquetas son mínimos. Estos indicadores sugieren que el modelo es consistente y confiable en contextos de clasificación multietiqueta.

La tabla de resultados presentada previamente ofrece una visión general del rendimiento del modelo en términos de métricas como precisión, recall y F1-score. Sin embargo, para comprender de manera más detallada cómo se distribuyen los aciertos y errores en cada clase, es necesario analizar las matrices de confusión.

En la matriz correspondiente a la clase cardiovascular (Figura 4 se observa un buen desempeño del modelo. De los casos realmente positivos, 247 fueron correctamente identificados, mientras que únicamente 13 se clasificaron de forma errónea como negativos. Por el lado de los negativos, 445 fueron reconocidos adecuadamente y solo 8 se confundieron como positivos. Estos resultados evidencian un equilibrio sólido entre sensibilidad y especificidad, lo que indica que el modelo logra diferenciar con bastante precisión los casos cardiovasculares de los que no lo son.

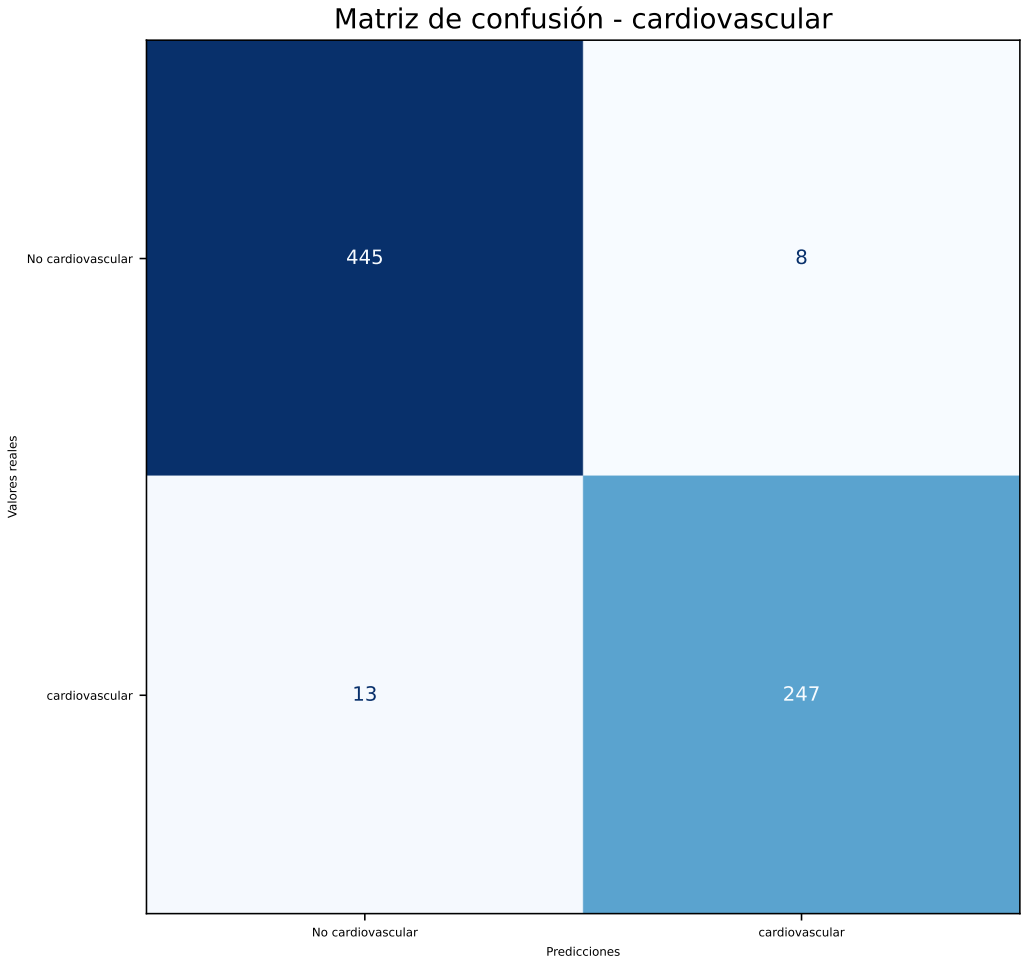


Figura 4: Matriz de confusión - Cardiovascular

En la clase hepatorenal (Figura 5, el modelo también presenta un rendimiento destacado. Se identificaron correctamente 216 positivos y 480 negativos, frente a solo 5 falsos positivos y 12 falsos negativos. Esto refleja un bajo nivel de error y una gran capacidad de discriminación. El modelo muestra aquí un comportamiento consistente, con una alta tasa de verdaderos positivos y un número reducido de predicciones incorrectas, lo que lo hace confiable para este tipo de diagnóstico.

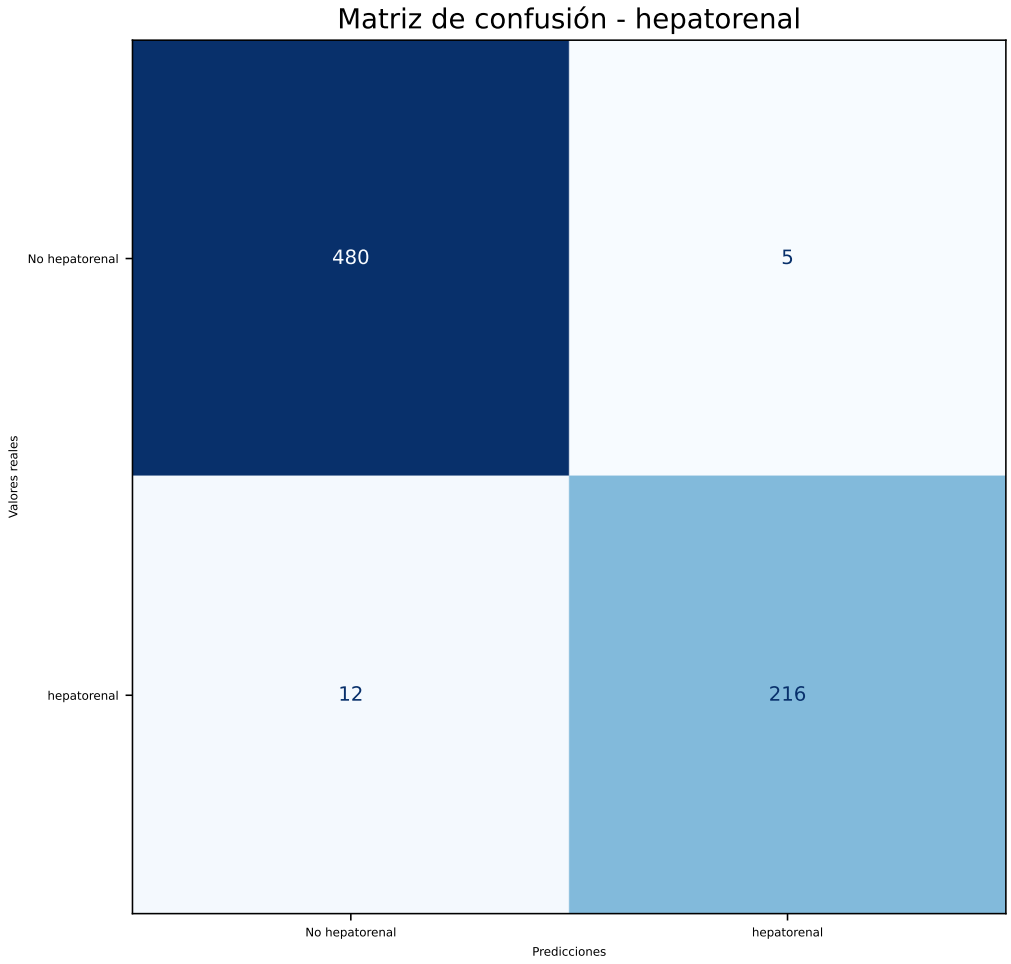


Figura 5: Matriz de confusión - Hepatorenal

En el caso de la clase neurológica (Figura 6 se percibe una mayor dificultad del modelo. Aunque se lograron clasificar correctamente 296 casos positivos y 343 negativos, se observa un número considerablemente más alto de errores: 32 falsos positivos y 42 falsos negativos. Esto implica que el modelo tiende a confundir con mayor frecuencia los casos neurológicos, afectando tanto la precisión como la sensibilidad. Por lo tanto, esta categoría representa un área de mejora importante para incrementar la robustez general del modelo.

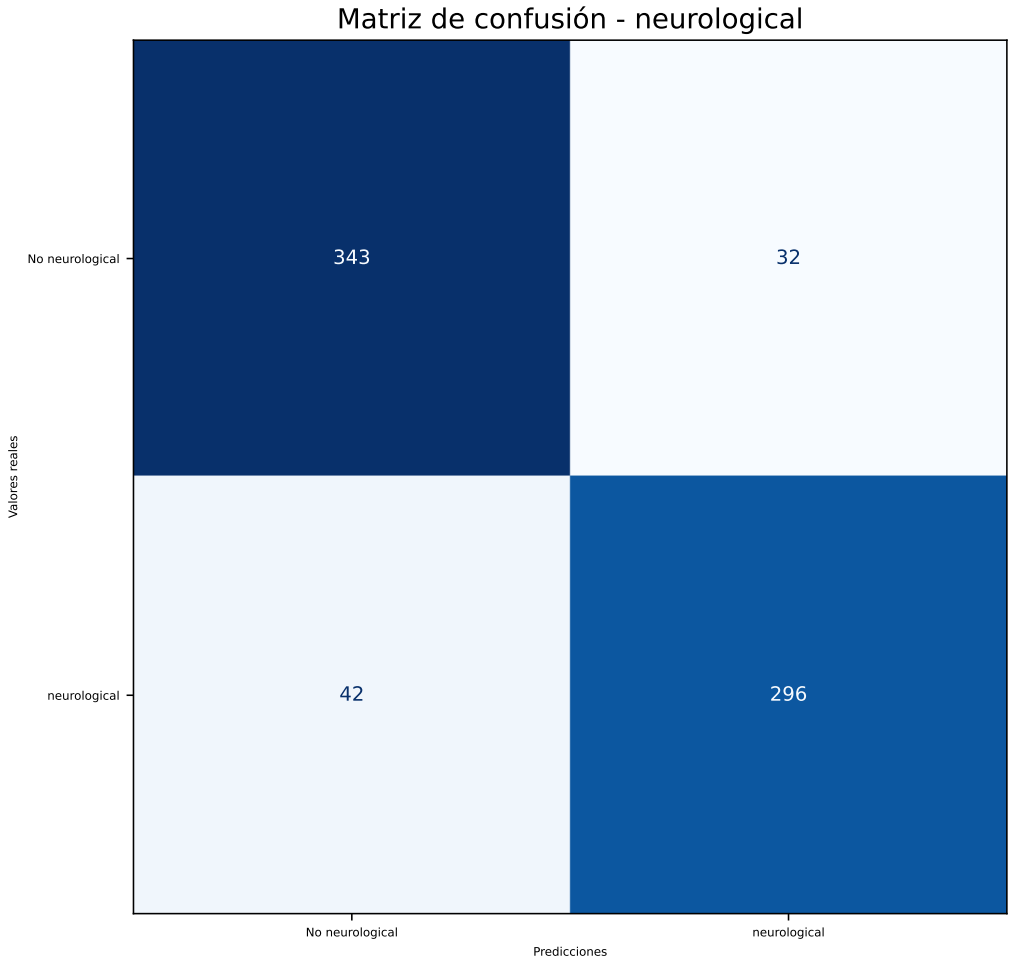


Figura 6: Matriz de confusión - Neurological

La matriz para la clase oncológica (Figura 9 muestra un desempeño muy favorable. El modelo identificó correctamente 124 positivos y 579 negativos, mientras que únicamente se produjeron 6 falsos negativos y 4 falsos positivos. Este comportamiento refleja una excelente capacidad de clasificación, con un balance óptimo entre la detección de verdaderos positivos y la minimización de errores. La alta precisión alcanzada en esta categoría sugiere que el modelo es especialmente confiable para la detección de casos oncológicos.

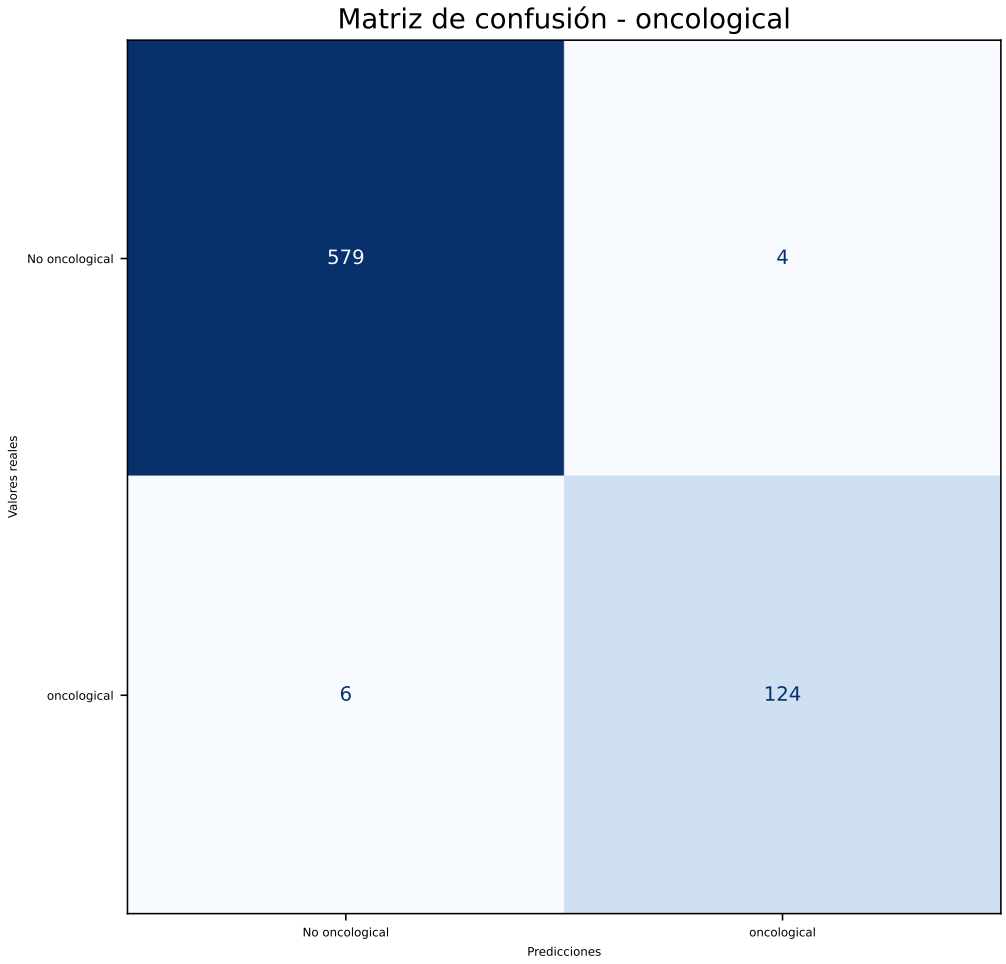


Figura 7: Matriz de confusión - Oncological

7. Visualización con V0

La imagen muestra la interfaz de v0.app, una plataforma impulsada por inteligencia artificial que permite a los usuarios crear aplicaciones web personalizadas mediante el uso de prompts en lenguaje natural. En este caso, el usuario solicita la creación de una aplicación frontend asociada a un proyecto de Machine Learning. Los requisitos específicos incluyen:

1. Un dashboard interactivo con métricas clave como F1-score y Accuracy.
2. Matriz de confusión visual para evaluar el rendimiento del modelo.
3. Gráficos de distribución de clases para visualizar cómo se agrupan los datos.

4. Una demo funcional para probar clasificaciones del modelo.
5. Visualización de las características más importantes del modelo.
6. Una interfaz gráfica para mostrar todos estos resultados de forma clara y accesible.

Una vez que se introduce este prompt, la plataforma genera automáticamente la estructura frontend de la aplicación, con la posibilidad de ajustarse a las necesidades del usuario, como cambiar colores o añadir funcionalidades adicionales. Además, la aplicación puede ser desplegada directamente desde la misma plataforma sin necesidad de escribir código adicional.

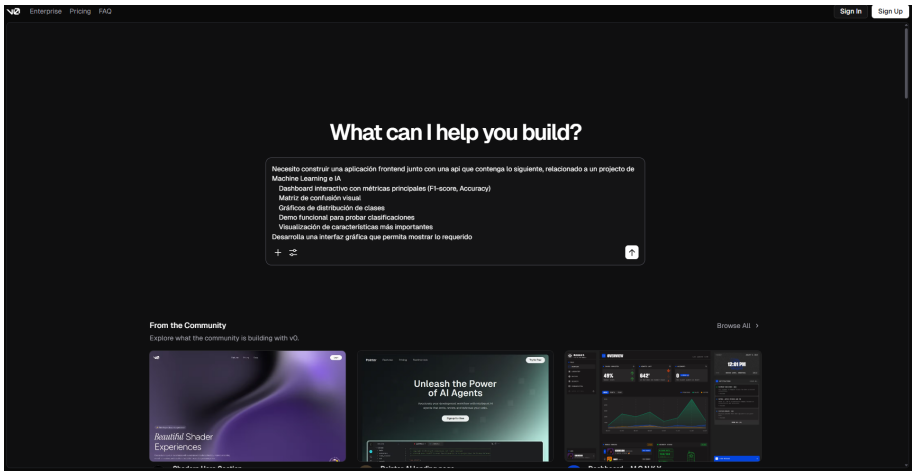


Figura 8: Prompt a V0 para la construcción de dashboard

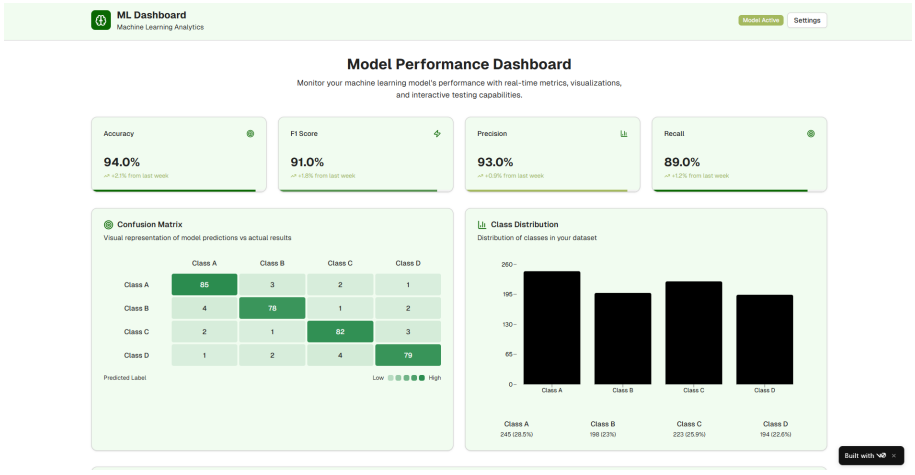


Figura 9: Visualización de dashboard con V0

8. Conclusiones

- El modelo utilizado demuestra un buen rendimiento en la mayoría de las categorías analizadas, alcanzando altos valores de acierto en las clases cardiovascular, hepatorenal y oncológica. Esto evidencia su capacidad para diferenciar entre casos positivos y negativos de manera confiable, lo que se refleja en un equilibrio favorable entre precisión y recall.

- La clase neurológica representa el mayor reto para el modelo, dado el incremento en la cantidad de falsos positivos y falsos negativos en comparación con las otras categorías. Esta limitación sugiere la necesidad de optimizar el modelo, ya sea mediante ajustes en los hiperparámetros, incorporación de más datos representativos de dicha clase o el uso de técnicas avanzadas de balanceo de datos.

- El análisis conjunto de las métricas globales y las matrices de confusión aporta una visión integral del comportamiento del modelo, permitiendo no solo validar su eficacia, sino también detectar patrones de error específicos. Esto brinda una base sólida para mejorar el sistema en futuras iteraciones, con miras a aplicaciones en contextos prácticos donde la clasificación correcta de cada patología es crítica.

Referencias

[1] Latam Republic. Tech sphere 2025 will bring together academia and industry around ai. URL: https://www.latamrepublic.com/tech-sphere-2025-will-bring-together-academia-and-industry-around-ai/?utm_source=chatgpt.com.

[2] Tech Sphere. Challenge de clasificación biomédica con ia. URL: https://techspherecolombia.com/ai-data-challenge/.